6. Basic I/O

5. Java and XML

Why XML?

- XML is a very useful technology for describing structured information
- XML tools make it easy to process and transform information
- XML has employed as the base language for communication protocols
- XML is widely used as protocol language in Java EE APIs

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <parent>
        <groupId>webapp.sample</groupId>
        <artifactId>web-parent</artifactId>
        <version>1.0-SNAPSHOT</version>
     </parent>
    <artifactId>web-app</artifactId>
    <packaging>jar</packaging>
    <name>Web Demo - Application UI project</name>
 </project>
```

What is an XML?

- **Extensible Markup Language** (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable
- It is a textual data format with strong support documents structure along with arbitrary data structures

The Structure of an XML Document

- An XML document should start with a **header** such as <?xml version="1.0"?> or <?xml version="1.0" encoding="UTF-8"?>

  A header is optional, but it is highly recommended
- The **body** of the XML document contains the root element (only one!), which can contain other elements (child elements)

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
     <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>webapp.sample</groupId>
        <artifactId>web-parent</artifactId>
        <version>1.0-SNAPSHOT</version>
     </parent>
    <artifactId>web-app</artifactId>
    <packaging>jar</packaging>
    <name>Web Demo - Application UI project</name>
 </project>
```

Element

- A logical document component either begins with a start-tag and ends with a matching end-tag or consists only of an empty-element tag:
  <modelVersion>4.0.0</modelVersion>

  <line-break />

Element (continued)

- An element can contain child elements, text, or both:

```
<parent>
        <groupId>webapp.sample</groupId>
        <artifactId>web-parent</artifactId>
        <version>1.0-SNAPSHOT</version>
   </parent>
```

Attributes

- A markup construct consisting of a name/value pair that exists within a start-tag or empty-element tag:

```
<project
xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
xsi:schemaLocation="http://maven.apache.org/P
OM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

Parsing an XML Document

- To process an XML document, you need to parse it:
  - read a file
  - confirm that the file has the correct format
  - break it up into the constituent elements
  - access those elements

Java XML Parsers

- Tree parser - Document Object Model (DOM) that read an XML document into a tree structure.
- Streaming parser - Simple API for XML (SAX) that generate events as they read an XML document.

XML namespace

- **XML namespaces** are used for providing uniquely named elements and attributes in an XML document
- A *namespace name* is a uniform resource identifier (URI)
- Typically, the URI chosen for the namespace of a given XML vocabulary describes a resource under the control of the author or organization defining the vocabulary

Namespace declaration

- An XML namespace is declared using the reserved XML attribute xmlns or xmlns:*prefix*, the value of which must be a valid namespace name:

  xmlns:xhtml="http://www.w3.org/1999/xhtml"
- Any element or attribute whose name starts with the prefix "xhtml:" is considered to be in the XHTML namespace

Default Namespace

- It is also possible to declare a default namespace:

  xmlns="http://www.w3.org/1999/xhtml"

- In this case, any element without a namespace prefix is considered to be in the XHTML namespace, if it or an ancestor has the above default namespace declaration
- Attributes are never subject to the default namespace

Well-formed XML document

- Well-formed = correct syntax
- The begin, end, and empty-element tags that delimit the elements are correctly nested, with none missing and none overlapping.
- The element tags are case-sensitive; the beginning and end tags must match exactly.
- There is a single "root" element that contains all the other elements

Valid XML Document

- Valid = well-formed + semantic-correct
- Semantic is described with:
  - Document Type Definition (DTD) or
  - XML Schema definition (XSD)
- Contains rules that explain how a document should be formed, by specifying the legal child elements and attributes for each element

Manuals

- http://docs.oracle.com/javase/tutorial/jaxp/index.html
- http://docs.oracle.com/javase/tutorial/jaxb/index.html